Package ‘genderizeR’

March 11, 2019

Type Package

Title Gender Prediction Based on First Names

Version 2.1.0

Description Utilizes the 'genderize.io' Application Programming Interface to predict gender from first names extracted from a text vector. The accuracy of prediction could be controlled by two parameters: counts of a first name in the database and probability of prediction.

License MIT + file LICENSE


BugReports https://github.com/kalimu/genderizeR/issues

Imports stringr (>= 1.0.0), httr (>= 1.1.0), tm (>= 0.6-2), data.table (>= 1.9.6), magrittr, parallel (>= 3.3.0), utils

Depends R (>= 3.3.0)

Encoding UTF-8

LazyData true

Suggests testthat, knitr, rmarkdown

RoxygenNote 6.1.1

VignetteBuilder knitr

NeedsCompilation no

Author Kamil Wais [aut, cre] (https://orcid.org/0000-0002-4062-055X), Nathan VanHoudnos [ctb], John Ramey [ctb], Thomas Klebel [ctb]

Maintainer Kamil Wais <kamil.wais@gmail.com>

Repository CRAN

Date/Publication 2019-03-11 07:32:38 UTC
**R topics documented:**

- authorships ......................................................... 2
- classificationErrors ........................................... 3
- findGivenNames .................................................. 4
- genderize ........................................................... 6
- genderizeAPI ....................................................... 7
- genderizeBootstrapError ........................................ 9
- genderizePredict .................................................. 10
- genderizeR .......................................................... 11
- genderizeTrain ..................................................... 11
- givenNamesDB_authorships ....................................... 13
- givenNamesDB_titles ............................................. 14
- numberOfNames .................................................... 14
- textPrepare ........................................................ 15
- titles ................................................................. 16

**Index**

<table>
<thead>
<tr>
<th>authorships</th>
<th>Authorships sample</th>
</tr>
</thead>
</table>

**Description**

A dataset containing a simple random sample of authorships (unique combination of authors and titles) from WebOfScience records of articles of "biographical-items" or "items-about-individual" types from all fields of study published from 1945 to 2014. The sample was drawn in December 2014.

**Usage**

authorships

**Format**

A data frame with 2641 rows and 5 variables:

- **title** The title of an article.
- **authors** All the authors of the article.
- **value** A single author of the article - with the title forms an authorship; there can be several authorships per article.
- **genderCoded** Manually coded gender of an author. There are four codes: "female", "male", "noname", "unknown". "Noname" is the code for a case were human coders were not able to find a first name of an author. "Unknown" is the code for a case were the coders found a full name of an author but were not able to verify if she or he is a man or a female.
- **WOSaccessionNumber** The original ID of an article in WebOfScience database.
classificationErrors

Source

http://webofknowledge.com/

classificationErrors  Calculating classification errors and other prediction indicators

Description

classificationErrors builds confusion matrix from manually coded and predicted gender vectors and returns classification errors calculated on that matrix.

Usage

classificationErrors(labels, predictions)

Arguments

labels  A vector of true labels. Should have following values: c("female", "male", "unknown", "noname"). noname stands also for initials only.
predictions  A vector of predicted gender. Should have following values: c("female", "male", NA). NA when it was not possible to predict a gender.

Value

A list of gender prediction efficiency indicators:

confMatrix  Full confusion matrix.
errorTotal  Total classification error calculated on the matrix.
errorFullFirstNames  Classification error calculated without "noname" category.
errorCoded  Classification error calculated without both "noname" and "unknown" category.
errorCodedWithoutNA  Classification error calculated only on "female" and "male" categories from both predictions and labels.
naTotal  Total proportion of items with unpredicted gender.
naFullFirstNames  Proportion of items with unpredicted gender calculated without "noname" category.
nacoded  Proportion of items with unpredicted gender calculated without both "noname" and "unknown" category.

errorGenderBias  Calculated as follows: "male" classified as "female" minus "female" classified as "male" and divided by the sum of items in "female" and "male" categories from both predictions and labels.
Examples

suppressWarnings(RNGversion("3.5.0"))
set.seed(23)
labels = sample(c("female", "male", "unknown", "noname"), 100, replace = TRUE)
predictions = sample(c("female", "male", NA), 100, replace = TRUE)
classificationErrors(labels, predictions)

   # $confMatrix
   # predictions
   # labels  female male <NA>
   # female  6  6  8
   # male  6 10 10
   # noname 12  6 17
   # unknown  5  7  7
   # <NA>  0  0  0
   #
   # $errorTotal
   # [1] 0.67
   #
   # $errorFullFirstNames
   # [1] 0.6461538
   #
   # $errorGenderBias
   # [1] 0

findGivenNames

Getting gender prediction data for a given text vector.

Description

findGivenNames extracts from text unique terms and predicts gender for them.
findGivenNames

Usage

findGivenNames(x, textPrepare = TRUE, country = NULL, language = NULL, apikey = NULL, queryLength = 10, progress = TRUE, ssl.verifypeer = TRUE)

Arguments

x A text vector or a character vector of unique terms pre-processed earlier manually or by the textPrepare function.
textPrepare If TRUE (default) the textPrepare function will be used on the x vector. Set it to FALSE if you already have prepared a character vector of cleaned up and deduplicated terms that you want to send to the API for gender checking.
apikey A character string with the API key obtained via https://store.genderize.io. A default is NULL, which uses the free API plan. If you reached the limit of the API you can start from the last checked term next time.
queryLength How much terms can be checked in a one single query.
progress If TRUE (default) progress bar is displayed in the console.
ssl.verifypeer Checks the SSL Certificate. Default is TRUE. You may set it to FALSE if you encounter some errors that break the connection with the API (though it is not recommended).

Value

A data table with given names found in database, gender predictions, probabilities of gender predictions, and counts how many people with a given name is recorded in the database.

Examples

x = "Tom did play hookey, and he had a very good time. He got back home barely in season to help Jim, the small colored boy, saw next-day's wood and split the kindlings before supper—at least he was there in time to tell his adventures to Jim while Jim did three-fourths of the work. Tom's younger brother (or rather half-brother) Sid was already through with his part of the work (picking up chips), for he was a quiet boy, and had no adventurous, trouble-some ways. While Tom was eating his supper, and stealing sugar as opportunity offered, Aunt Polly asked him questions that were full of guile, and very deep—for she wanted to trap him into damaging revealments. Like many other simple-hearted souls, it was her pet vanity to believe she was endowed with a talent for dark and mysterious diplomacy, and she loved to contemplate her"
most transparent devices as marvels of low cunning.
(from 'Tom Sawyer' by Mark Twain)"

xProcessed = textPrepare(x)

foundNames = findGivenNames(xProcessed, textPrepare = FALSE)
foundNames[count > 100]

# (the results can differ due to new, updated data pulled from the API)
# name gender probability count
# 1:   jim male  1.00   2291
# 2:   mark male  1.00   6178
# 3:  polly female  0.99    191
# 4:   tom male  1.00   3736

# localization
findGivenNames("andrea", country = "us")
# name gender probability count
# 1: andrea female  0.97   2308

findGivenNames("andrea", country = "it")
# name gender probability count
# 1: andrea male  0.99  1070

genderize

Description

For each character string in a x vector genderize function using an output of the findGivenNames function and returns a gender prediction for the whole character string based on first names located inside the strings.

Usage

genderize(x, genderDB, blacklist = NULL, progress = TRUE)

Arguments

x A vector of text strings.
genderDB A data table output of findGivenNames function for the vector x.
blacklist A character vector of terms (stopwords) that will be excluded from gender checking.
progress If TRUE (default) progress bar is displayed in the console.
Value

A data table with text string, a term found in genderDB, that is finally used as a given name to predict gender of the string, a predicted gender, a number of potential gender indicators (eg. 1 if only one term from the text string is found in genderDB).

Examples

```r
x = c("Winston J. Durant, ASHP past president, dies at 84",
"Gold Badge of Honour of the DGAI Prof. Dr. med. Norbert R. Roewer Wuerzburg",
"The contribution of professor Yu.S. Martynov (1921-2008) to Russian neurology",
"JAN BASZKIEWICZ (3 JANUARY 1930 - 27 JANUARY 2011) IN MEMORIAM",
"Maria Sklodowska-Curie")

givenNames = findGivenNames(x)
givenNames = givenNames[count>40]
genderize(x, genderDB=givenNames, blacklist=c('med'))

# text
# 1: Winston J. Durant, ASHP past president, dies at 84
# 2: Gold Badge of Honour of the DGAI Prof. Dr. med. Norbert R. Roewer Wuerzburg
# 3: The contribution of professor Yu.S. Martynov (1921-2008) to Russian neurology
# 4: JAN BASZKIEWICZ (3 JANUARY 1930 - 27 JANUARY 2011) IN MEMORIAM
# 5: Maria Sklodowska-Curie

givenName gender genderIndicators
# 1: winston  male  1
# 2: norbert  male  1
# 3: yu       female 1
# 4: jan      male  1
# 5: maria    female 1
```

Description

The genderizeAPI function connects to genderize.io API and checks if a term (one or more) is in the genderize.io database and returns predicted gender probability and count of the records with this term in the database.

Usage

```r
genderizeAPI(x, country = NULL, language = NULL, apikey = NULL,
             ssl.verifypeer = TRUE)
```
Arguments

x A vector of terms to check in genderize.io database.
apikey A character string with the API key obtained via https://store.genderize.io. A default is NULL, which uses the free API plan. If you reached the limit of the API you can start from the last checked term next time.
ssl.verifypeer If TRUE (default) it checks the SSL Certificate.

Value

A list of four elements: response is a data frame with names, genders, probabilities and counts or NULL if no terms are found in the genderize.io database; limitLeft is showing how many API queries are still possible within the current limit which will be renewed in limitReset seconds.

Examples

## Not run:

terms = c("loremipsum")
genderizeAPI(terms)$response
# Null data.table (0 rows and 0 cols)

terms = c("jan", "maria", "norbert", "winston", "loremipsum")
genderizeAPI(terms)

# example of the function output
$response
name gender probability count
1:  jan male 0.60 1692
2:  maria female 0.99 8467
3:  norbert male 1.00 77
4:  winston male 0.98 128

$limitLeft
[1] 967

$limit
[1] 1000

$limitReset
[1] 83234

## End(Not run)
Gender prediction errors on bootstrap samples

description

genderizeBootstrapError calculates the Apparent Error Rate, the Leave-One-Out bootstrap error rate, and the .632+ error rate from Efron and Tibishirani (1997). The code is modified version of several functions from sortinghat package by John A. Ramey.

usage

genderizeBootstrapError(x, y, givenNamesDB, probs, counts, num_bootstraps = 50, parallel = FALSE)

arguments

  x  A text vector that we want to genderize
  y  A text vector of true gender labels ('female' or 'male') for x vector
  givenNamesDB  A dataset with gender data (could be an output of findGivenNames function)
  probs  A numeric vector of different probability values. Used to subsetting a givenNamesDB dataset
  counts  A numeric vector of different count values. Used to subsetting a givenNamesDB dataset
  num_bootstraps  Number of bootstrap samples. Default is 50.
  parallel  It is passed to genderizeTrain function. If TRUE it computes errors with the use of parallel package and available cores. Default is FALSE.

value

A list of bootstrap errors:

  apparent  Apparent Error Rate
  loo_boot  LOO-Boot Error Rate
  errorRate632plus  .632+ Error Rate

see also

In the sortinghat package.
Examples

## Not run:

```r
x <- c('Alex', 'Darrell', 'Kale', 'Lee', 'Robin', 'Terry', rep('Robin', 20))
y <- c(rep('female', 6), rep('male', 20))
givenNamesDB = findGivenNames(x)
pred = genderize(x, givenNamesDB)
classificationErrors(labels = y, predictions = pred$gender)

probs = seq(from = 0.5, to = 0.9, by = 0.05)
counts = c(1)
set.seed(23)
genderizeBootstrapError(x = x, y = y,
givenNamesDB = givenNamesDB, probs = probs, counts = counts,
num_bootstraps = 20, parallel = TRUE)
```

# $apparent
# [1] 0.9615385

# $loo_boot
# [1] 0.965812

# $errorRate632plus
# [1] 0.964225

## End(Not run)

genderizePredict        Gender predicting function

Description

The `genderizePredict` function predicts gender using the values of probability and count parameters that minimize the error coded.

Usage

`genderizePredict(trainedParams, newdata, givenNamesDB)`
genderizeR

Arguments

- `trainedParams` An output of a `genderizeTrain` function with prediction efficiency indicators for different combinations of probability and count values.
- `newdata` A character vector for gender prediction.
- `givenNamesDB` A dataset with gender data (could be an output of `findGivenNames` function).

Value

A character vector of values: `male`, `female` or `unknown`.

Description

The `genderizeR` package uses genderize.io API to predict gender from first names extracted from text corpuses. The accuracy of prediction could be controlled by two parameters: counts of first names in database and probability of gender given the first name.

Details

If you need help with your research or commercial projects, feel free to contact me via my homepage contact form: https://kalimu.github.io/

See Also

- https://kalimu.github.io/project/genderizer/ [R package homepage]
- https://github.com/kalimu/genderizeR [source code of the latest development version of the R package]
- http://genderize.io/ [homepage of genderize.io API]

genderizeTrain

Training `genderize` function

Description

The `genderizeTrain` function predicts gender and checks different combinations of probability and count parameters.

Usage

`genderizeTrain(x, y, givenNamesDB, probs, counts, parallel = FALSE, cores = NULL)`
Arguments

- **x** A text vector that we want to genderize.
- **y** A text vector of true gender labels for the x vector.
- **givenNamesDB** A dataset with gender data (could be an output of `findGivenNames` function).
- **probs** A numeric vector of different probability values. Used to subsetting a givenNamesDB dataset.
- **counts** A numeric vector of different count values. Used to subsetting a givenNamesDB dataset.
- **parallel** If TRUE it computes errors with the use of parallel package and available cores. Default is FALSE.
- **cores** A integer value for number of cores designated to parallel processing or NULL (default). If parallel argument is TRUE and cores is NULL, than the available number of cores will be detected automatically.

Value

A data frame with prediction indicators for each combination of parameters:

- **errorcoded** The classification error for predicted and unpredicted gender.
- **errorcodedwithoutna** The classification error for items with predicted gender only.
- **nacoded** The proportion of items with manually coded gender and with unpredicted gender.
- **errorgenderbias** The net gender bias error.

See Also


Examples

```r
## Not run:

x = c('Alex', 'Darrell', 'Kale', 'Lee', 'Robin', 'Terry', 'John', 'Tom')
y = c(rep('male', length(x)))

givenNamesDB = findGivenNames(x)
probs = seq(from = 0.5, to = 0.9, by = 0.1)
counts = c(1, 10)

genderizeTrain(x = x, y = y, 
    givenNamesDB = givenNamesDB,
    probs = probs, counts = counts,
    parallel = TRUE)

# prob count errorcoded errorcodedwithoutna nacoded errorgenderbias
```
givenNamesDB_authorships

### Gender data for authorship sample

#### Description

A dataset with first names and gender data from genderize.io for the `authorships` dataset in the package. This is the output of the `findGivenNames` function that was performed on December 26, 2014.

#### Usage

`givenNamesDB_authorships`

#### Format

A data.table object with 872 rows and 4 variables:

- **name**: A term used as a given name.
- **gender**: The predicted gender for the term.
- **probability**: The probability of the predicted gender.
- **count**: How many social profiles with the term as a given name are recorded in the genderize.io database.

#### Source

http://genderize.io/
### givenNamesDB_titles

**Gender data for titles sample**

**Description**

A dataset with a gender data from genderize.io for the `titles` dataset in the package. This is the output of `findGivenNames` function that was performed on December 26, 2014.

**Usage**

```r
givenNamesDB_titles
```

**Format**

A data.table object with 872 rows and 4 variables:

- **name**: A term used as first name.
- **gender**: The predicted gender for the term.
- **probability**: The probability of the predicted gender.
- **count**: How many social profiles with the term as a given name is recorded in the genderize.io database.

**Source**

[http://genderize.io/](http://genderize.io/)

### numberOfNames

**Number of names in the database.**

**Description**

`numberOfNames` returns a number of distinct names in the genderize.io database scrapped from genderize.io page.

**Usage**

```r
numberOfNames()
```

**Value**

returns a numeric value
textPrepare

Examples

numberOfNames()

| textPrepare | Preparing text vector for gender prediction |

Description

The `textPrepare` function takes a text vector as an argument and converts it into a vector of unique terms. This function is used by default by the `findGivenNames` function as a text pre-processor before sending a query to the genderize.io API.

Usage

`textPrepare(x, textPrepMessages = FALSE)`

Arguments

- `x` A vector of character strings.
- `textPrepMessages` If TRUE verbose output of the preparing process is shown on the console (default is FALSE).

Value

A vector of unique terms with at least two characters.

Examples

```r
x = c("Winston J. Durant, ASHP past president, dies at 84",
      "Gold Badge of Honour of the DGAI Prof. Dr. med. Norbert R. Roewer Wuerzburg",
      "The contribution of professor Yu.S. Martynov (1921-2008) to Russian neurology",
      "JAN BASZKIEWICZ (3 JANUARY 1930 - 27 JANUARY 2011) IN MEMORIAM",
      "Maria Sklodowska-Curie")

head(textPrepare(x))
```
Description

A dataset containing a simple random sample of article titles from WebOfScience records of articles of "biographical-items" or "items-about-individual" types from all fields of study published from 1945 to 2014. The sample was drawn in December 2014.

Usage
titles

Format

A data frame with 1190 rows and 2 variables:

- **title**: The title of an article.
- **genderCoded**: Manually coded gender of a person mentioned in the title. There are four codes: "female", "male", "both", "none". "None" is the code for a case where human coders were not able to find a full name in the title or verify if she or he is a man or a female. "Both" is the code for two rare cases in the dataset where two people were mentioned in the title and one of them was male and the other was female.

Source

http://webofknowledge.com/
Index

*Topic datasets
  authorships, 2
  givenNamesDB_authorships, 13
  givenNamesDB_titles, 14
titles, 16

authorships, 2

classificationErrors, 3

findGivenNames, 4
genderize, 6
genderizeAPI, 7
genderizeBootstrapError, 9
genderizePredict, 10
genderizeR, 11
genderizeR-package (genderizeR), 11
genderizeTrain, 11
givenNamesDB_authorships, 13
givenNamesDB_titles, 14

numberOfNames, 14
textPrepare, 15
titles, 16